

Article

Deep Learning for Estimating the Fill-Level of Industrial Waste Containers of Metal Scrap: A Case Study of a Copper Tube Plant

Kosmas Alexopoulos ^{1,*}, Paolo Catti ¹, Giannis Kanellopoulos ¹, Nikolaos Nikolakis ¹, Athanasios Blatsiotis ², Konstantinos Christodoulopoulos ², Apostolos Kaimenopoulos ² and Efstathia Ziata ²

¹ Laboratory for Manufacturing Systems & Automation (LMS), Department of Mechanical Engineering & Aeronautics, University of Patras, 26504 Patras, Greece

² Halcor, Copper & Alloys Extrusion Division of ElvalHalcor S.A., 32011 Oinofyta, Greece

* Correspondence: alexokos@lms.mech.upatras.gr; Tel.: +30-2610910160

Abstract: Advanced digital solutions are increasingly introduced into manufacturing systems to make them more intelligent. Intelligent Waste Management Systems in industries allow for data collection and analysis to make better-informed decisions, monitor and manage processes remotely, and improve waste management. In many industries, scrap is collected in large waste containers located on the factory floor, usually close to its source. In most cases, monitoring of waste containers' fill levels is either manually performed by visual inspection by the operators working in close proximity or by employing intrusive mechanical systems such as weight sensors. This work presents a computer vision system that uses Deep Learning (DL) and Convolutional Neural Network (CNN) for the automated estimation of the fill level in industrial waste containers of metal scrap. The training method and parameters as well as the classification performance of VGG16 CNN that was retrained upon images collected in the field, are presented in detail. The proposed method has been validated upon an industrial case study from the copper tube production industry in which the fill level of two waste containers is estimated. A total of 9772 images were captured for the first container and 11,234 images for the second container. The VGG16 model achieved an accuracy from 77.5% to 95% on the testing dataset. The industrial case study demonstrates that the proposed computer vision system has sufficient accuracy for classifying the fill levels of metal scrap containers which allows for the development of waste management applications in industrial environments.

Keywords: Industry 4.0; digitalization of industry; scrap management; convolutional neural networks



Citation: Alexopoulos, K.; Catti, P.; Kanellopoulos, G.; Nikolakis, N.; Blatsiotis, A.; Christodoulopoulos, K.; Kaimenopoulos, A.; Ziata, E. Deep Learning for Estimating the Fill-Level of Industrial Waste Containers of Metal Scrap: A Case Study of a Copper Tube Plant. *Appl. Sci.* **2023**, *13*, 2575. <https://doi.org/10.3390/app13042575>

Academic Editor: Alexandre Carvalho

Received: 30 December 2022

Revised: 12 February 2023

Accepted: 13 February 2023

Published: 16 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The importance of maintaining the production of manufacturing systems within the desired boundaries of costs quality and time, while preserving certain levels of flexibility for addressing unplanned situations had a great impact on manufacturing research receiving particular attention [1]. Under this perspective, industrial processes could greatly benefit from the integration of Industry4.0 digital solutions [2] and Artificial Intelligence (AI) technologies [3,4] that can be used for the management of activities on a factory floor [5]. Among other activities, optimized management of scrap, generated during production, contributes to lean manufacturing, reducing production deadlock situations that may occur due to poor waste management. For example, the lack of space in waste containers may result in unplanned delays in the production process.

The main contribution of this work is a computer vision system that uses Deep Learning (DL) and Convolutional Neural Network (CNN) for the automated estimation of the fill level in industrial waste containers of metal scrap. The system has been implemented and deployed in an industrial environment for copper products.

This work consists of 5 chapters. Chapter 2 reviews the scientific literature for relevant topics including Industry 4.0, computer vision methods with a focus on machine and

deep learning architectures, and solutions and wastebins' fill-level approaches. Chapter 3 presents the main steps of the proposed method. Chapter 4 details the application of the aforementioned method in an industrial case of a copper tube plant. Finally, Chapter 5 discusses the results and presents future directions. The industrial case study demonstrates that the proposed method has sufficient accuracy that allows its practical application in industrial environments for estimating the fill-level of industrial waste containers. The fill-level information can be further exploited for managing scrap on the manufacturing floor.

In the metal industries, scrap usually consists of recyclable materials like aluminum, iron, and copper, left over after manufacturing processes. Its value, however, remains high as it can be collected and recycled. A concept for an intelligent waste management system in the copper industry is presented in [6]. It considers the use of an Industrial Internet of Things (IIoT) platform for receiving and storing waste data from the production, to identify abnormalities and/or deviations from pre-defined thresholds. Moreover, scrap is collected in waste containers, whose management becomes important if not efficiently handled during production. Filled-up waste containers may interrupt production until they are either emptied or replaced. Thus, the monitoring and estimation of the fill level become important for preserving a smooth production. This will allow companies to manage their scrap collection process more efficiently, something that can reduce operational costs and environmental impact. This process can be supported by the use of sensors and computer vision methods [7]. The decreasing costs of the Internet of Things (IoT) are trending nowadays, and it can be applied in a wide range of fields. Waste management using IoT has a very good prospect and is required for reducing the growing pollution in the environment [8]. In [9], sonar sensors have been used to monitor wastebins in smart cities to provide measurements from 2 cm to 400 cm, with a 3 mm accuracy. In [10], they used two ultrasonic sensors inside waste bins for collection and monitoring of the level of waste inside the bin. In [11], they deployed infrared and sonar-based sensors for metal waste monitoring in a manufacturing site.

Machine and computer vision has gathered intensive academic attention and the use of this technology has spread rapidly, because of its advantages. Machine vision technology with human-like vision capability has revolutionized the process of automated inspection [12,13]. An image-based framework considering pre-trained CNN, ResNet-101 to detect surface defects during the centerless grinding of tapered rollers has been developed in [14]. Similarly, neural networks were used for identifying the dimensional patterns and classifying the profiles on a rubber weather-strip extrusion production line in [15]. Computer vision systems using cameras which take a real-time picture of the bin and analyse its fill level have also been used for waste monitoring. A bin level detection model based on the grey level co-occurrence matrix feature extraction approach has been developed in [16]. In [17], they presented a system for the automation of waste auditing in industrialized construction facilities in which the waste generated during the cutting process was quantified using contour-based image processing algorithms, and the identification of the material was determined by deep learning classification models. A Smart Bin with computer vision and IoT that can separate waste automatically with a ResNet-50 Deep Learning model has been developed for classifying and separating different types of waste [18].

To the best of our knowledge, no work has been reported that aims at classifying the fill-level of metal scrap containers based on computer vision and deep learning technologies and then integrating the computer vision system into an information platform for managing the information. This work aims to describe and implement such a system and evaluate it in a case study of a copper tube plant.

2. Computer Vision and Deep Learning Method for Industrial Scrap Containers Fill-Level Estimation

Based on the results of the state-of-the-art analysis, three main options were identified for industrial scrap containers fill-level estimation. Computer vision systems, ultrasonic sensors, and sensors can measure weight. Ultrasonic sensors are commonly used in

industrial environments for measuring the fill level of waste bins and other applications (e.g., [10]). However, the installation of an ultrasonic sensor, in some cases such as the case studied in this work, is not efficient in terms of not disturbing the production processes in a factory. The sensor should have been placed within or above the bin container at an inclined angle. Such a placement would have introduced high risk of the sensor being damaged during the bin filling as well as the bin collection processes. This risk would have required for sensor mounting and unmounting procedures during these processes which is not efficient. Weight sensors for relatively large containers may be considered an expensive and intrusive solution to the working environment. These sensors may also have considerable error and provide inaccurate results in estimating the bin fill-level. Different scrap loads may have the same weight but take up different volumes in the container, because of different load patterns, thus corresponding to significantly different bin fill levels. The cost and maintenance of a vision system for industrial waste bin fill level estimation depends on several factors, such as the type of cameras used, the complexity of image processing algorithms, and the number of cameras required for the application. In this work, a low-cost machine vision system was developed as low-cost area scan cameras, using a software triggering mechanism and no frame grabber hardware. These systems can provide accurate results about the fill level and can be less affected by factors such as the fill pattern of the scrap in the bin, and are not intrusive to the working environment.

This work presents a novel computer vision system that can automatically recognize the fill level of industrial waste containers with metal scrap. The fill pattern may differ in terms of waste location within a container, geometry and shape of the waste, and relative position compared to the optical sensor (camera). The proposed approach can classify the fill level in different categories by capturing RGB images of the waste containers and applying them as input to a pre-trained CNN model that executes the classification task. The results of the fill-level estimation process are managed by an IIoT platform and visualized in a user-friendly web application. In Figure 1 the flow diagram of the proposed method is presented.

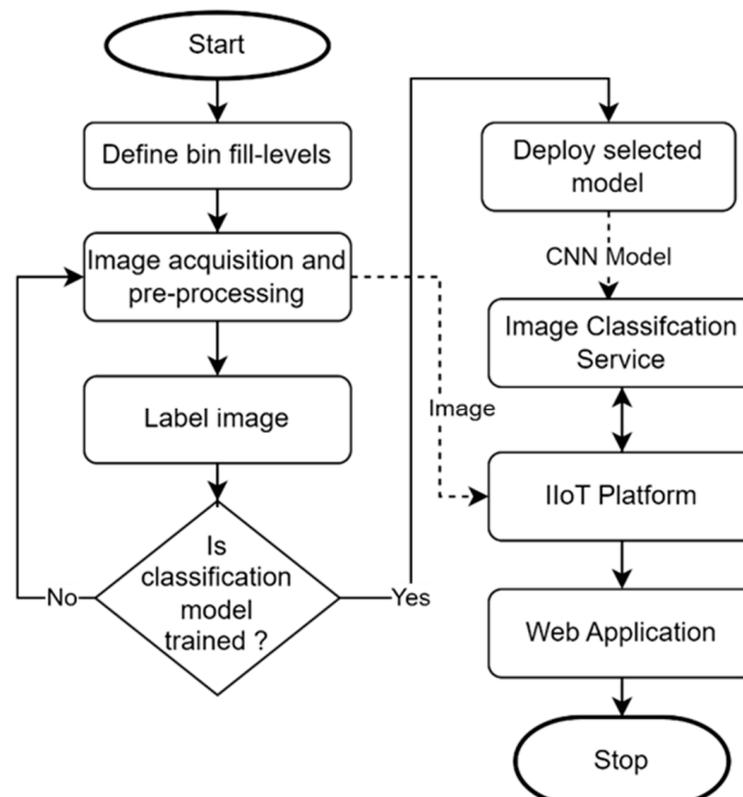


Figure 1. Flow-diagram of the proposed method.

1. *Define bin fill levels:* This step decides on the definition of the fill levels of the waste bins. The fill level ranges from 0% (empty bin) to more than 100% (overfilled bin). Moreover, as a special class, the “no-bin” class can be included, signifying the absence of the container within the field of view (e.g., the container was previously moved by a forklift). The granularity of the fill-levels from 0% to 100% is defined by taking into account field experts’ capability to classify the fill level of a container as well, as the level of accuracy needed by the estimation model for offering waste management services on top of the fill-level assessment. In this work, as explained in the industrial case description, a step of 10% has been adopted so that the levels were (1–10%), (11–20%) etc. These levels are then used for labelling captured images for model training purposes.
2. *Image data acquisition and pre-processing:* Images are collected by optical sensors. Pre-processing of image data takes place to make sure that the image fits the needs of the classification task. In some cases, cameras may serve multiple purposes and are not used solely for fill-level monitoring. For example, they may be used for tracking the presence or absence of some specific equipment in the area. In such cases, the images could be cropped around the area of interest.
3. *Image labelling (classification):* In case the CNN classification model needs training, the images captured are labelled by an expert (human) in one of the fill-level classes, defined in step 1. The labelled images are then provided as input to the training step that follows.
4. *Training and evaluation of CNN model:* By the time a sufficient number of images is collected, open-source DL models for image classification, such as VGG16, that are already trained in huge image datasets such as ImageNet [19] are customized and retrained on the labelled images. The captured images are resized for fitting to the input dimension of the CNN models (in the industrial pilot of this study a 224×224 resizing of the images took place). The accuracy and loss of the CNN model during training and evaluation are calculated. Accuracy depicts the number of correct model predictions of the overall predictions while loss provides the difference between the expected outcome and the outcome produced by the CNN model. The training accuracy represents the accuracy of the model at fitting the data. Validation accuracy represents an unbiased metric on how well the model fits the training data while finetuning the model’s hyperparameters. The accuracy is calculated based on the following Formula (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where:

TP: True Positive,
 TN: True Negative,
 FP: False Positive,
 FN: False Negative.

The loss is calculated based on the used loss function. In all cases, the used loss function was categorical cross-entropy. Categorical cross-entropy is given by Formula (2) below:

$$L = \sum_{i=1}^M y_i \log(\hat{y}_i), \quad (2)$$

where:

y is a vector of entries for each class and \hat{y}_i is the vector of predictions that contains the probabilities for each outcome, which needs to sum to 1.

5. *IIoT platform, image classification service and web application:* The trained model is deployed within an Image Classification Service that runs on the services layer of a digital IIoT platform. The layered architecture proposed in [2] and adapted in [6]

has been adopted in this work for integrating the different layers from factory-level edge devices (cameras) to IIoT platforms and Web applications. The adopted layered architecture allows for structured deployment of software components and hardware devices both at the edge and the cloud, while at the same time the integration among the different layers is based on clearly defined roles and interfaces. Images captured by the computer vision system are populated through IoT protocols such as Message Queuing Telemetry Transport (MQTT) or Representational State Transfer Application Programming Interface (REST API) function calls to the IIoT platform. Through a publish-subscribe mechanism, new images trigger the call of the Image Classification Service that classifies each new image to one of the fill levels and stores this information back to the IoT platform. A web application is linked to the IIoT platform for providing end-user functionality such as dashboards with bin-level statuses and trendlines.

3. Industrial Case Study

3.1. Industrial Pilot

The method presented in Chapter 2 has been applied in an industrial pilot in the production line of a copper tube plant. In Figure 2, the process plan of the copper tube production under study is presented. The process starts with copper billets being preheated before entering the hot extrusion press to produce the “mother” tube. These tubes, which have an initial length of 15 to 30 m, are then reduced in diameter and wall thickness by either drawing or cold rolling processes. Further reductions are then performed at Breakdown lines and Spinner Blocks by drawing the tube further with successive passes, till the final or semi-final dimension is reached. In the spoolers’ workstations, the copper pipe is formed in coils, to be forwarded for annealing to the Bright Annealing Furnaces and then for insulation with polyethylene foam. The final step of the production is the packaging and storing of tubes.

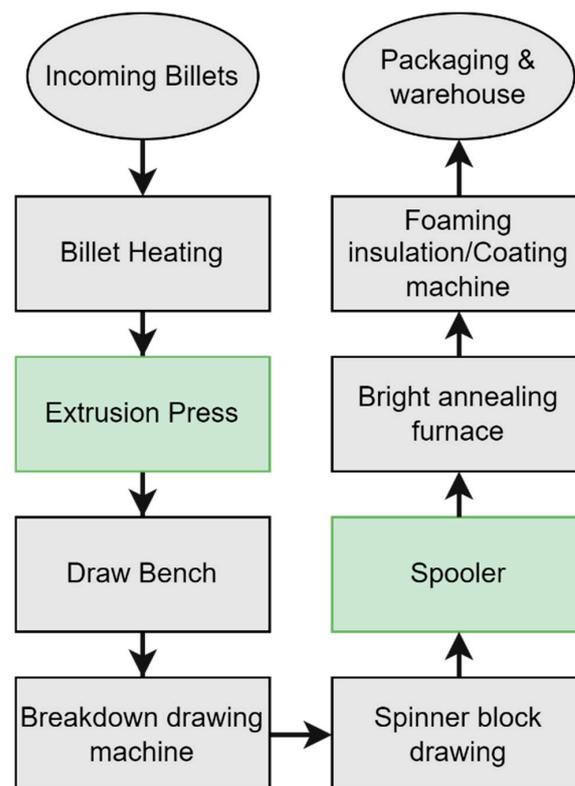


Figure 2. Copper Tube Plant process plan.

The workstation where the proposed solution has been applied is the Spoolers' workstation, due to the amount of scrap these machines are producing. More specifically, the Spoolers produce level wound coil (semi-) final products and the scrap generated is collected in containers. A container may be used to collect scrap for a group of Spooler machines. In Figure 3 samples of the bins located close to the Spoolers are depicted.

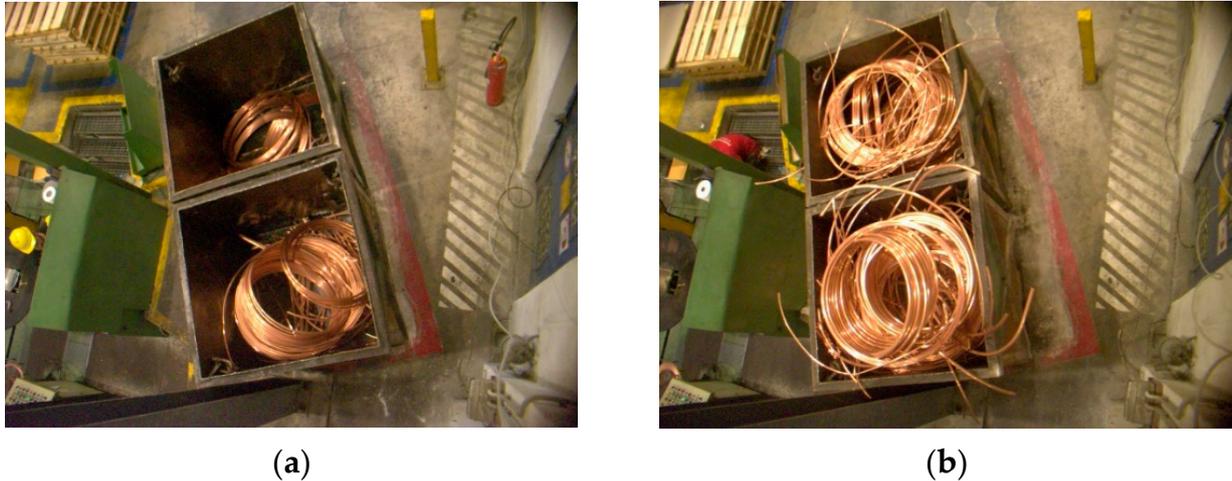


Figure 3. Waste bins (a) Spoolers' scrap 20% fill-in level for the bottom bin and (b) Spoolers' scrap 100% fill-in level for bottom bin.

3.2. Pilot System Configuration and Deployment

To support the industrial scenario, within the pilot environment described above, the system presented in Figure 4 has been deployed in the pilot area of the spoolers' workstation. For the implementation of the computer vision system, one area scan, GigE interface, and camera were used (Table 1). A consistent setup was utilized both during the training and testing phase of the images. The pilot case involved the placement of a camera approximately 2 m above the waste bins, in close proximity to a wall, on a stable iron structure, utilizing a camera mount bracket with a 360/90 degree rotational capability. The camera was connected to a computer machine via an ethernet cable for the purpose of triggering the software and storing the captured images. The location of the waste bin remained constant in accordance with factory regulations, and the bins were always situated within a specific frame formed on the floor, with an acceptable deviation of approximately 10 cm.

The following software modules have been developed and deployed (Figure 4):

- *Camera Control App*: A python application that is used to control, configure, manage, and trigger the camera for capturing images in a pre-set periodic manner (one image per five minutes). The Camera Control App receives the captured images and then stores them into the IIoT platform through REST API calls and the publish-subscribe mechanism.
- *IIoT platform*: An extension of the IIoT platform developed in Java and initially presented in [2] has been used for the needs of this pilot. The platform has been extended to implement the Asset Administration Shell (AAS) model [20] for storing and managing information collected from sensors in the shop-floor or legacy systems.
- *DL Classification App*: This application subscribes to image publishing events and when a new image is published by the Camera Control App it applies the classification model and returns the result, in JSON format, to the IIoT platform. The application is developed in Python and uses Tensorflow ML library. The development of the DL model is described in detail in chapter 3.3.

- *Legacy Systems Connector*: In order to provide a holistic insight into the status of the shop floor, the information provided by the vision system is combined with additional information regarding the status of the activities on the shop floor and is loaded to the IIoT system.
- *Waste Management Web Application*: The waste management web application presents the bin fill level and different types of Key Performance Indicators in a dashboard (Figure 5) for monitoring purposes by the engineers.

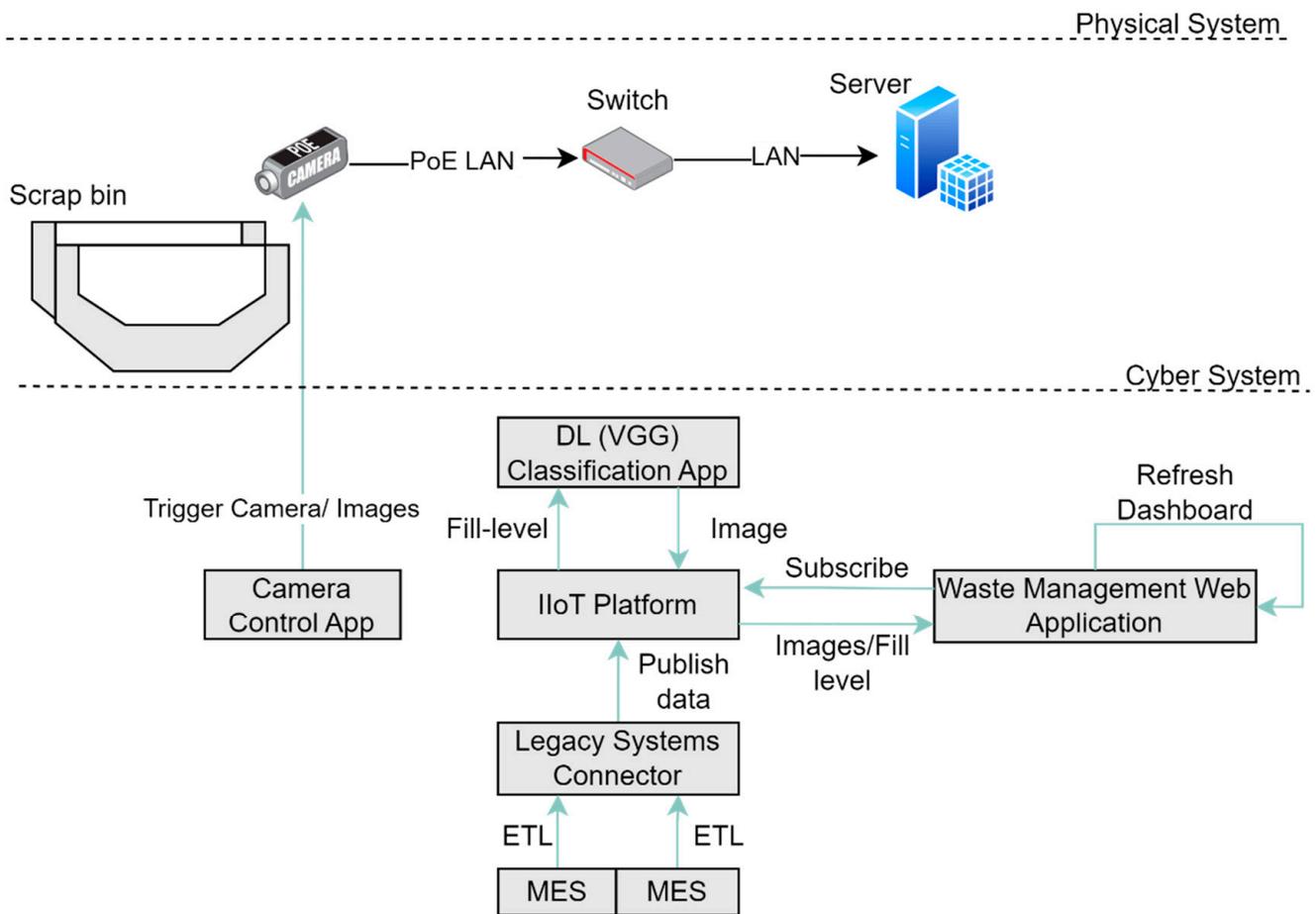
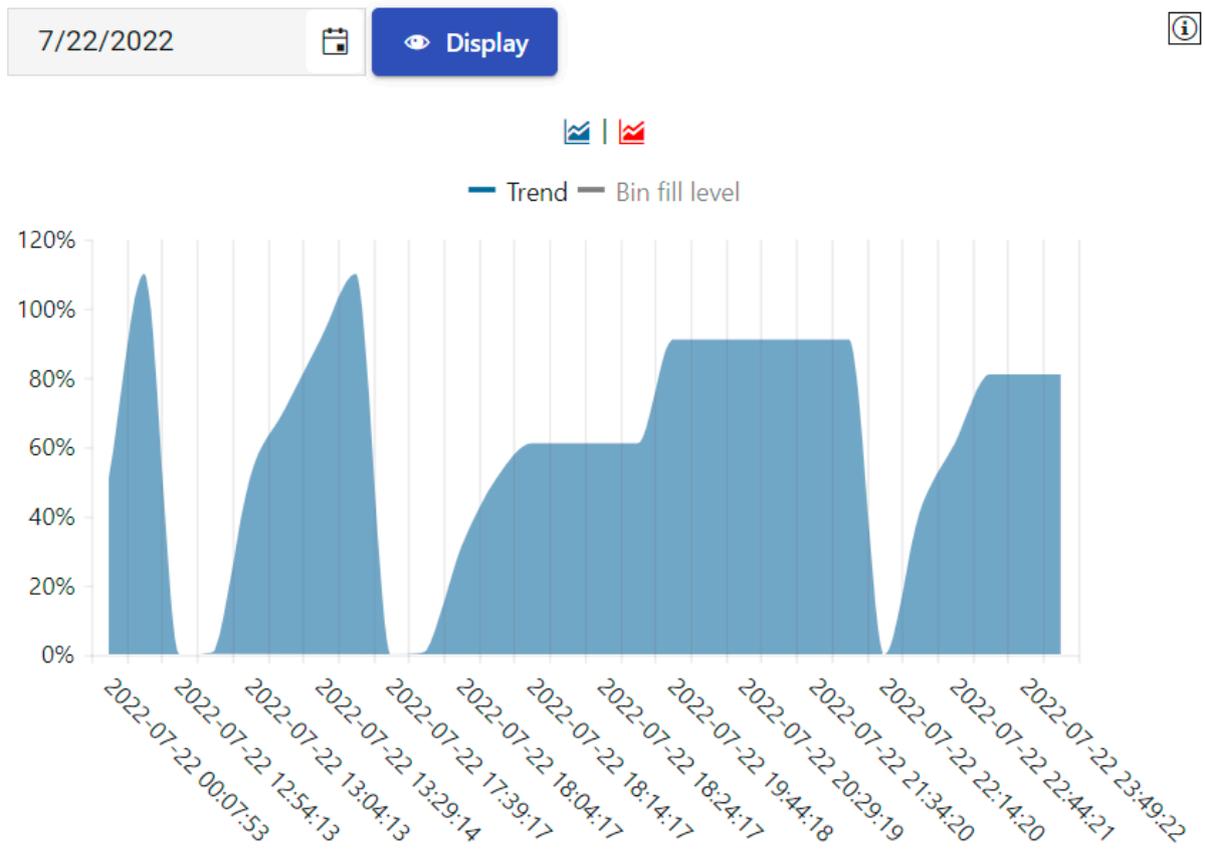


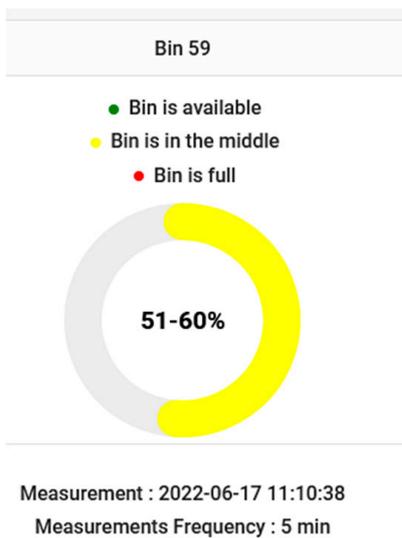
Figure 4. System architecture for supporting the industrial pilot case.

Table 1. Camera specifications.

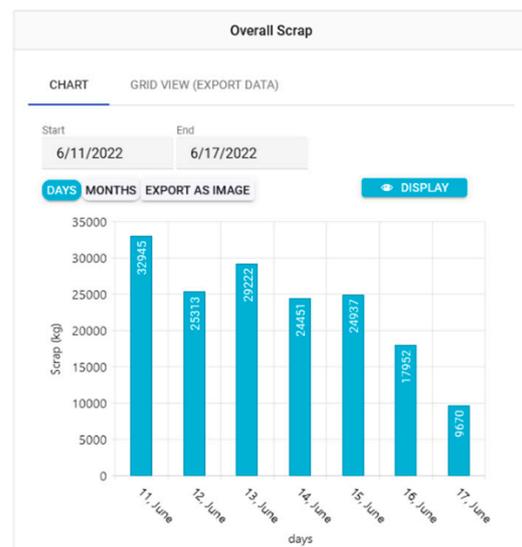
| Attribute | Value |
|-----------------------|--------------------------|
| Sensor type | CMOS |
| Resolution | 1600 px × 1200 px (2 MP) |
| Frame Rate | 60 FPS |
| Mono/Color | Color |
| Interface | GigE |
| Operating temperature | 0–50 °C |
| Power Consumption PoE | 2.7 W |



(a)



(b)



(c)

Figure 5. Visualization components of the dashboard: (a) Bin fill level over time, (b) Current bin-fill level, and (c) Scrap generation over time.

3.3. Machine Learning Model Development

In this chapter, the implementation of the method, described in Chapter 3, upon the industrial pilot case, is presented.

- *Define bin fill levels:* For the purpose of the case study in this work, twelve (12) fill-levels were defined as follows: 0% (empty bin), [1–10%], [11–20%], [21–30%], [31–40%], [41–50%], [51–60%], [61–70%], [71–80%], [81–90%], [91–100%], and greater than 100% (“Overfilled”), and a class which signify the absence of a bin (“No Bin”).
- *Image data acquisition and pre-processing:* The first step is using the camera to capture an image of the bin. During the time of the study, more than twenty thousand images have been captured (Table 2). A sample picture which is captured from the camera is shown in Figure 3. To build the training dataset, each image needs to be pre-processed. Image pre-processing starts by cropping the top and bottom parts and the next step is to rotate the image clockwise by 90 degrees. After the rotation, the picture is then cut down the middle into two parts creating two images, one for each bin (see Figure 6). The bins are always placed inside a red rectangle drawn on the floor. This means that the placement of the bins is fixed but can deviate a few centimetres in all directions. Since one camera was used to capture the images for both bins, due to the camera position not all details of Bin 2 can be captured. Due to the position and field of view of the camera, there is a blind spot on the left side of Bin 2 (see Figure 6b) which can make the fill level recognition more challenging for this particular bin. The placement of the camera was based on the needs of the manufacturer for giving priority to monitoring Bin 1.
- *Image classification (labelling):* After discussion with users and operators, the classification/labelling of the captured images took place. Images were classified according to the fill level of the bin shown. Labelling was performed by examining the captured images one by one and placing them in the appropriate class inside the dataset. This was done for both Bin 1 and Bin 2. In Figure 7 sample images for both Bin1 and Bin2 are provided.

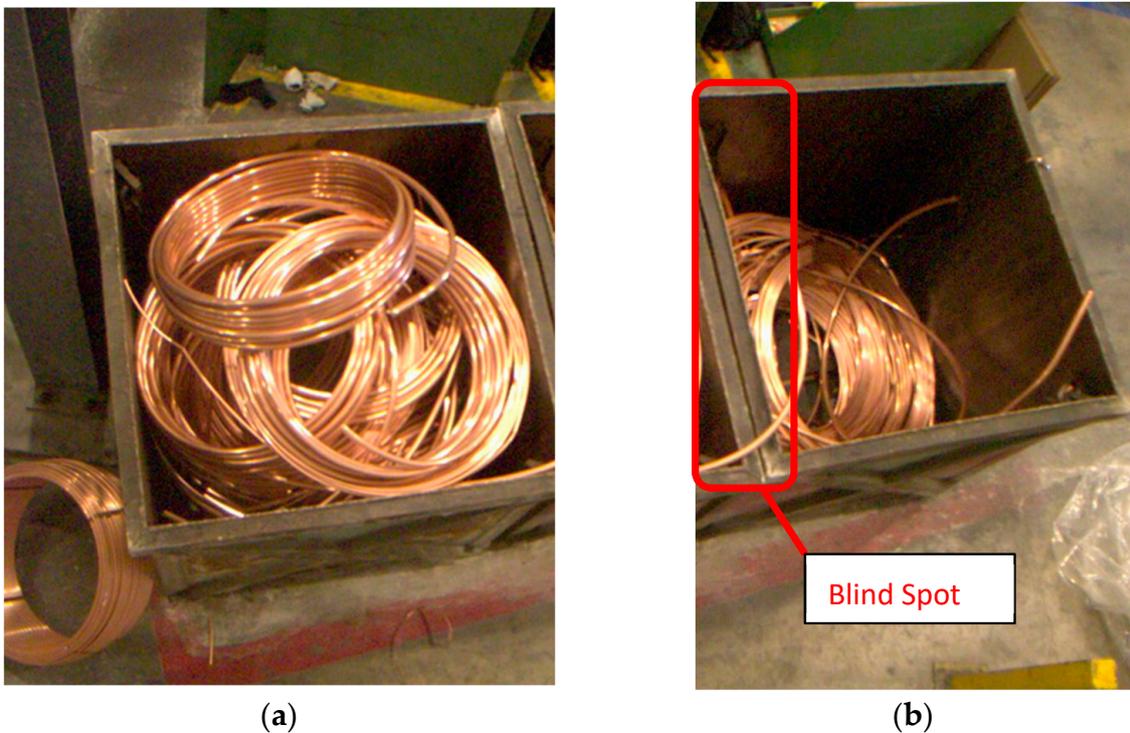


Figure 6. (a) Image of Bin1 after pre-processing, (b) Image of Bin2 after pre-processing; the blind spot is indicated.

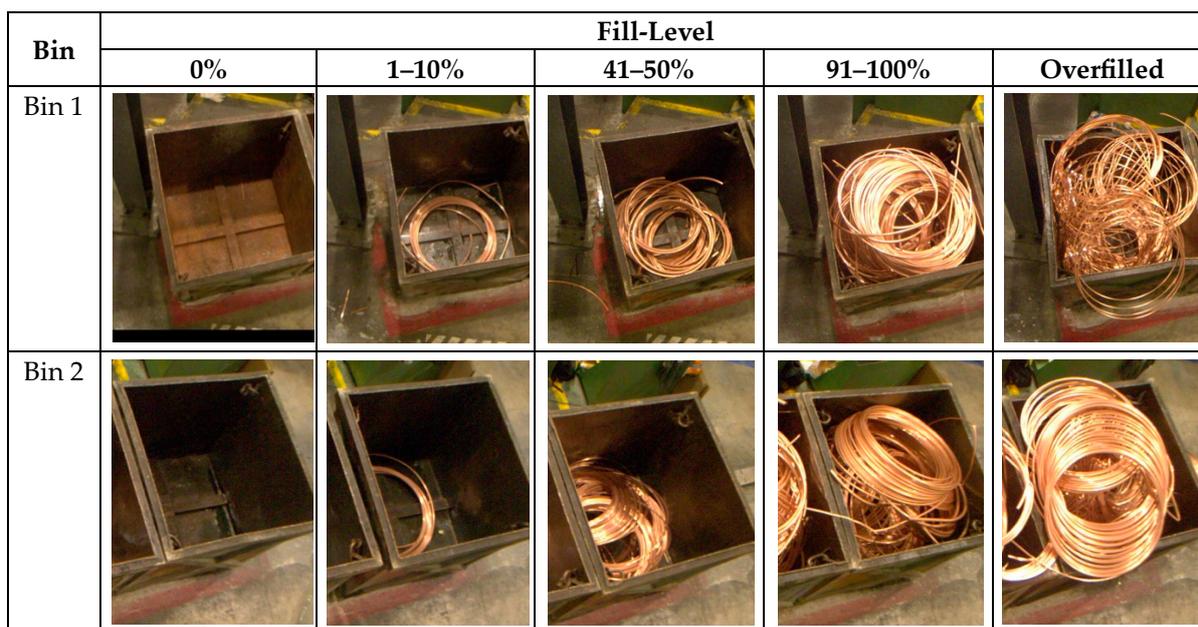


Figure 7. Samples of different fill levels for Bin 1 and Bin 2.

An annotated dataset of 9772 images for Bin 1 and 11,234 images for Bin 2 was created. In Table 2 there is an overview of the number of photos labelled per bin and fill level.

Table 2. Labelled data per bin and fill-level.

| Bin | Fill Levels (%) | | | | | | | | | | | | No Bin | Overfilled |
|-----|-----------------|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|----|--------|------------|
| | 0 | 1-10 | 11-20 | 21-30 | 31-40 | 41-50 | 51-60 | 61-70 | 71-80 | 81-90 | 91-100 | | | |
| 1 | 236 | 466 | 245 | 441 | 846 | 1101 | 600 | 1514 | 838 | 1803 | 885 | 12 | 784 | |
| 2 | 2087 | 2087 | 932 | 632 | 729 | 1058 | 1009 | 963 | 719 | 440 | 422 | 13 | 142 | |

- *Training and evaluation of CNN model:* The VGG16 [21] image classifier CNN model, pre-trained on the ImageNet benchmark dataset, was selected for executing the image classification task. By using a pre-trained model with initial weight values, less images can be used for training, rather than starting from scratch. This reduces the amount of time needed for image acquisition and labelling as well as the required computational power to train the model. A customized model for each bin based on the initial model was developed using the Tensorflow Keras [22]. The following training setup was used:
 - The annotated dataset of images (Table 2) was utilized for training.
 - For both Bin 1 and Bin 2, the split for training, validation, and testing of the dataset in Table 2 was 70%, 20%, and 10% respectively.
 - The images were normalized, and their input shape set to $224 \times 224 \times 3$.
 - The batch size for the training was set to 32.
 - The number of training epochs used was 12 for the model for Bin 1 and 29 for the model for Bin 2. The number of epochs was not statically determined at the beginning of the training process. Instead, an early stopping function was used which monitored the validation loss. The function was implemented with a patience of 3, which restored the best weights. For Bin 1, the validation loss threshold after which training stopped was 0.19. For Bin 2, the validation loss threshold after which training stopped was 0.14.
 - The values of the network weights were initialized to ImageNet ones (pre-trained to ImageNet dataset).
 - Pooling was set to “avg”, meaning that Global Average Pooling was applied to the output of the last convolutional block.

- All network layers were un-frozen and free to be retrained on the new data.
- The top, dense, layer available by default in Tensorflow Keras was replaced with a custom one that works for classifying 13 classes. The new top layer consisted of the following consecutive layers and the model’s architecture can be seen in Figure 8:
 - a. A Flatten Layer
 - b. A Dense Layer with 4096 units and its activation function set to “relu”
 - c. A Dropout layer with a rate of 0.4
 - d. A Dense Layer with 1024 units and its activation function set to “relu”
 - e. A last Dense Layer with 13 units and its activation function set to “softmax”
- To optimize the model parameters, the Adam [23] optimizer was used in all CNN models. The learning rate was set at 0.00001.



Figure 8. Architecture of customized VGG16 model.

After training was completed, the customized VGG16 model achieved the following performance results:

- For Bin 1, the model achieved 96% training accuracy, 93% validation accuracy, and 89% accuracy on the testing set. The model’s loss was 0.1.
- For Bin 2, the model achieved 96% accuracy on the training set, 94% validation accuracy, and 95% accuracy on the testing dataset. The model’s loss was 0.14.

In Figure 9, the graphs for the accuracy and loss performance of VGG16 model in Bin 1 are shown.

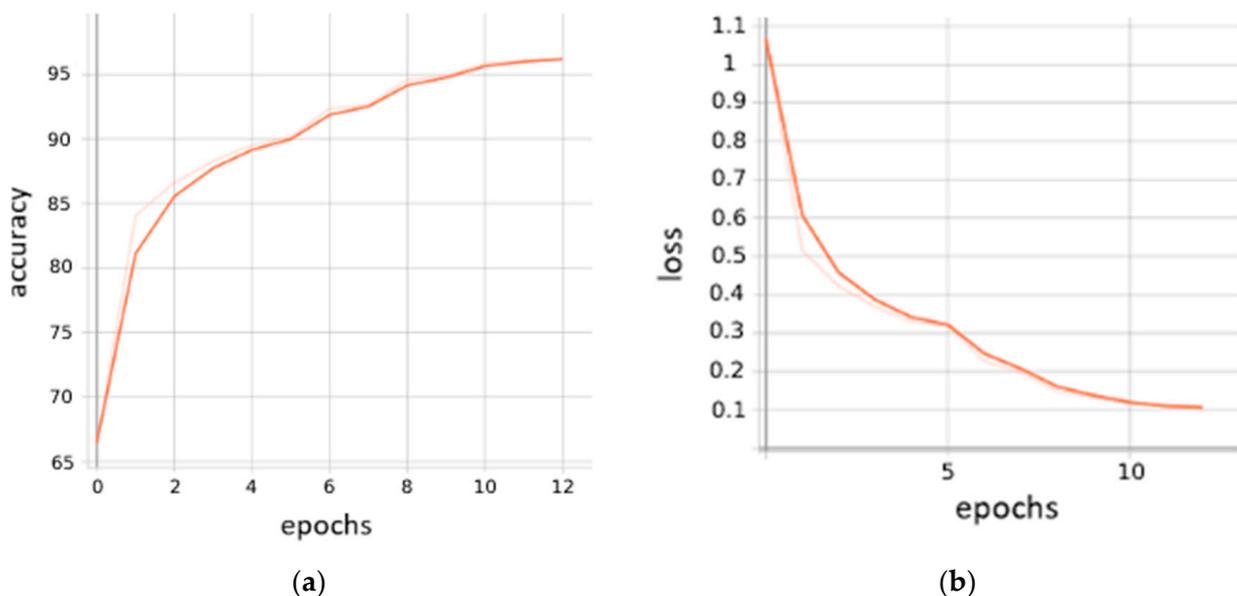


Figure 9. Performance of customized VGG16 model for Bin 1: (a) model accuracy and (b) model loss.

In Figures 10 and 11, the graphs for the accuracy and loss performance of the VGG16 model in Bin 2 are presented.

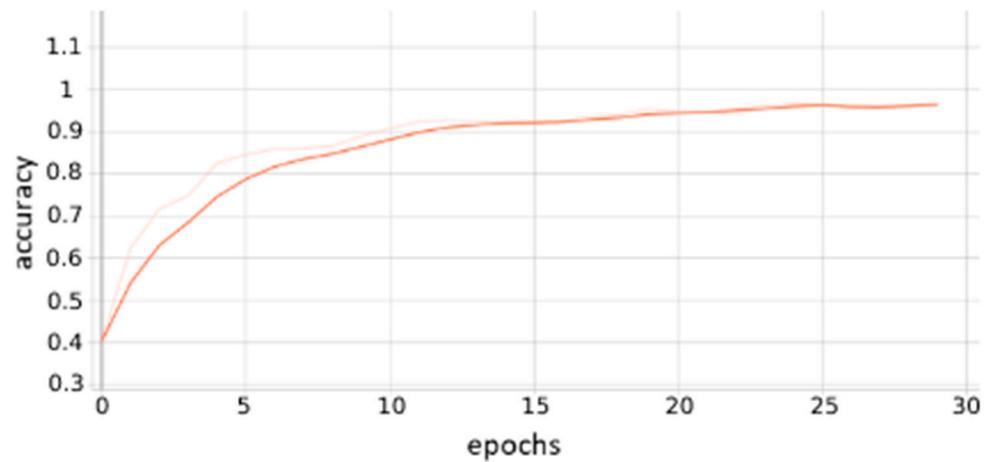


Figure 10. Accuracy of customized VGG16 model for Bin 2.

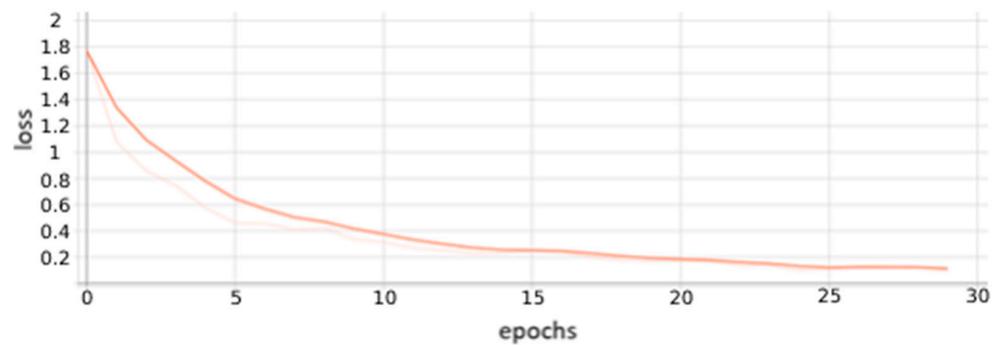


Figure 11. Loss of customized VGG16 model for Bin 2.

In order to assess the performance of the customized VGG16 model, three additional state of the art, off-the-shelf image classifiers were tested upon the developed dataset. These models include the ResNet50 model [24], the InceptionV3 [25], and the MobileNetV2 model [26]. In Table 3, the results of the training, evaluation, and testing of each model are presented.

Table 3. Comparison between customized VGG16, ResNet50, InceptionV3, and MobileNetV2 CNN models when tested for Bin 1 and Bin 2.

| | VGG16 | ResNet50 | InceptionV3 | MobileNetV2 |
|-------|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Bin 1 | Train. Accuracy: 96% Val. Accuracy: 93% Loss: 0.1 Test. Accuracy: 89% | Train. Accuracy: 99% Val. Accuracy: 97% Loss: 0.2 Test. Accuracy: 82% | Train. Accuracy: 98% Val. Accuracy: 95% Loss: 0.22 Test. Accuracy: 81% | Train. Accuracy: 75% Val. Accuracy: 71% Loss: 0.63 Test. Accuracy: 72% |
| Notes | Best Performing Model | Overfitting | Overfitting | Worst Performing Model |
| Bin 2 | Train. Accuracy: 96% Val. Accuracy: 94% Loss: 0.14 Test. Accuracy: 95% | Train. Accuracy: 95% Val. Accuracy: 93% Loss: 0.2 Test. Accuracy: 92% | Train. Accuracy: 88% Val. Accuracy: 91% Loss: 0.28 Test. Accuracy: 88% | Train. Accuracy: 69% Val. Accuracy: 73% Test. Accuracy: 71% |
| Notes | Best Performing Model | Overfitting | Underfitting | Worst Performing Model—Underfitting |

After the training was completed, the VGG16 CNN model was tested with 884 new pictures (which were not present in the training, validation or the previous testing set) taken from Bin1. These new images were used to validate the previous score of 89% accuracy on the first testing set and get a more accurate picture of the model’s performance on new data. Of these 884 pictures, 800 were correctly classified while 84 pictures were misclassified. This results in an accuracy of 90.5%, similar to that achieved on the first testing set. In Table 4, we can see a detailed look at the exact misclassification (false-positive) percentage for each class.

Table 4. Analysis of the bin 1 images that were wrongly classified by the customized VGG16 model.

| | No Bin | 0% | 1–10% | 11–20% | 21–30% | 31–40% | 41–50% | 51–60% | 61–70% | 71–80% | 81–90% | 91–100% | Overfilled | Total |
|--------------------------------|--------|----|-------|--------|--------|--------|--------|--------|--------|--------|--------|---------|------------|-------|
| Total Images Misclassified | 4 | 0 | 0 | 9 | 7 | 11 | 28 | 4 | 5 | 2 | 4 | 10 | 0 | 84 |
| % Of Misclassified Images | 4.7% | 0% | 0% | 10.7% | 8.3% | 13.0% | 33.8% | 4.7% | 5.9% | 2.3% | 4.7% | 11.9% | 0% | 100% |
| % Relatively to the Entire Set | 0.45% | 0% | 0% | 1.01% | 0.79% | 1.24% | 3.17% | 0.45% | 0.56% | 0.27% | 0.45% | 1.13% | 0% | 9.5% |

On further analysis, precision, recall, F1-score, and the weighted average were calculated for the Bin 1 model and the results can be found below (Table 5).

Table 5. Evaluation indexes of customized VGG16 Bin 1 model.

| Precision | Recall | F1-Score | Weighted Average |
|-----------|--------|----------|------------------|
| 0.9044 | 0.9153 | 0.9098 | 0.9184 |

As a next step, the delta of the misclassification for these 84 images was analyzed. Out of the 84 images, 53 were misclassified as one class above or below the class they truly belonged to. This is shown in Table 6.

Table 6. Analysis of the delta difference between the calculated fill level and the actual one for the customized VGG16 model in bin 1.

| Delta of Misclassification | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Total |
|----------------------------------------------|-------|-------|-------|------|---|---|---|---|---|----|----|----|-------|
| Sum of Misclassified Images | 53 | 15 | 9 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 84 |
| % Relatively to the Entire Misclassified Set | 63.2% | 17.8% | 10.7% | 8.3% | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% |

Similarly, the CNN model was tested with new images for Bin 2. In this case, 200 new captured images (which were not present in the training, the validation, nor the previous testing set) were used to test the model. These new images were tested to validate the score on the first testing set and get a full picture of the model’s performance on new data. A total of 155 out of the 200 images were correctly classified, while 45 were not. This resulted in an accuracy of 77.5%. In Table 7, a detailed look at the exact misclassification percentage for each class is presented.

Table 7. Analysis of the bin 2 images that were wrongly classified by the customized VGG16 model.

| | No Bin | 0% | 1–10% | 11–20% | 21–30% | 31–40% | 41–50% | 51–60% | 61–70% | 71–80% | 81–90% | 91–100% | Overfilled | Total |
|--------------------------------|--------|----|-------|--------|--------|--------|--------|--------|--------|--------|--------|---------|------------|-------|
| Total Images Misclassified | 0 | 0 | 0 | 8 | 22 | 2 | 2 | 0 | 6 | 0 | 2 | 1 | 2 | 45 |
| % Of Misclassified Images | 0% | 0% | 0% | 17.9% | 48.9% | 4.4% | 4.4% | 0% | 13.4% | 0% | 4.4% | 2.2% | 4.4% | 100% |
| % Relatively to the Entire Set | 0% | 0% | 0% | 4% | 11% | 1% | 1% | 0% | 3% | 0% | 1% | 0.5% | 1% | 22.5% |

Upon further investigation, precision, recall, F1-score, and weighted average were calculated for the Bin 2 model and the results can be found below (Table 8).

Table 8. Evaluation indexes of customized VGG16 Bin 2 model.

| Precision | Recall | F1-Score | Weighted Average |
|-----------|--------|----------|------------------|
| 0.7743 | 0.7745 | 0.7744 | 0.8307 |

Similar to Bin 1, the delta of the misclassification for the 45 images was calculated. Out of the 45 pictures, 16 were misclassified 2 classes above or below the class they truly belonged to, 14 had a delta of 3, 11 had a delta of 1, and 4 were 4 classes above or below the actual class. This is shown in Table 9.

Table 9. Analysis of the delta difference between the calculated fill level and the actual one for the customized VGG16 model in Bin 2.

| Delta Of Misclassification | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | TOTAL |
|----------------------------------------------|-------|-------|-------|------|----|----|----|----|----|----|----|----|-------|
| Sum of Misclassified Images | 11 | 16 | 14 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 45 |
| % Relatively to the Entire Misclassified Set | 24.5% | 35.5% | 31.1% | 8.9% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% |

The scores of the evaluation of the model for Bin 2 differ by 17.5% and the 200 testing images were qualitatively investigated. A closer observation of the pictures showed that these pictures were significantly different from the ones used for the training and validation stage. For example, it was observed that in the second testing dataset there was a significant number of images where the scrap was placed inside the bin in a different vertical orientation compared to the horizontal placements, seen during the training phase, where Bin-2 is filled more uniformly. Due to the specific use of Bin 2 from the factory operators, who use it mainly for storing scrap after Bin 1 is filled in, the variability of the pictures captured and eventually used during the training stage might not be representative. In order to validate that the model of Bin 2 can learn from new images, a subset of these 200 images was used to re-train the model, which was then tested and found to be significantly improved. Consequently, for the Bin 2 classification model to become more generalizable, the model should be trained with data containing more variability (e.g., recorded on different days and times).

4. Conclusions and Future Work

This work presents a computer vision system along with its integration into an IIoT platform for identifying and classifying the fill levels of waste containers used for collecting metal scrap in manufacturing industries. Deep Learning models and more specifically the VGG16 CNN model has been trained for performing the classification task in a case study deriving from a copper tube plant. The industrial case study demonstrates that the VGG16 model has sufficient accuracy for classifying the fill levels of metal scrap containers which allows their practical application in industrial environments. In the context of the industrial pilot case, two VGG16 CNN models have been trained for two different metal scrap containers. The models have been trained with the batch size parameter set to 32, with approximately 9.000 annotated images for one model and approximately 11.000 images for the second model. They achieved an accuracy ranging from 77.5% to 95%, depending on the model and on the testing dataset. Moreover, the work also presents an implemented IIoT-based system to manage the information generated by the computer vision system and communicate it to the industrial users via the IIoT information stack. The proposed method can be implemented using standard, relatively low-cost, off-the-shelf hardware and software components for computer vision systems, especially for cases in which other sensor-based systems cannot be applied.

The current work did not investigate the performance of the model based on different types of waste in the bins. However, it can be expected that the trained model would be a good starting point that can be further retrained with additional images that would, however, be considerably less in volume compared to the ones used in the current work, thus saving costs from model development task. One shortcoming of the proposed method is that two distinct models were trained for each of the two bins of the industrial pilot case. Due to the intrinsic differences in the camera field of view between bin 1 and bin 2, depicted in the images (see Figure 7), the pictures between the two bins differ significantly. In order to have achieve a common, generalized model, mounting of a second camera would be required, which would provide a similar field of view to bin 2, similar to the existing field of view for bin 1, which, however, would require additional costs that would not be accepted by the factory management, given the fact that bin 2 does not have the same impact on the operation of the work center as bin 1. However, investigating the generalization of the CNN model will be a topic of future research.

The images were labelled by experts who visually estimated the fill level. However, due to the geometry of the copper tube waste, their estimates could be biased, as the visual difference in 10% increments could be small, and as a result, similar images might be labelled to different classes. The issue of mislabelled images can be responsible for some of the model misclassifications. In the future, in order to address this issue, work will take place in the direction of annotating the images by automatically adding reference lines, that indicate the height of the bin at different points, that could support the homogenization of the labelling. The user will label the images with regards to these reference lines and thus misclassifications will be reduced. Moreover, future work will focus transfer the models developed for one workstation to other workstations of the production line, in which scrap may have different properties such as geometry or bin filling patterns. This would alleviate some of the training spent in labelling images for the different fill-in levels. Moreover, to further alleviate the tedious, costly, and error-prone process of manual labelling, automated methods based on synthetic datasets [27] can also be considered for undertaking this task. Finally, based on the data collected, fill-level prediction models will be developed that can be used for extending the services offered from fill-level visibility to prediction that can further improve the waste management practices applied.

Author Contributions: Conceptualization, K.A.; funding acquisition: A.K.; investigation: A.B. and K.C.; methodology, K.A., P.C. and N.N.; project administration: E.Z.; software, P.C. and G.K.; supervision, K.A.; validation, E.Z., K.C., A.B. and A.K.; visualization: P.C., G.K. and N.N., writing—review and editing, K.A., P.C., G.K., N.N., K.C., A.B., A.K. and E.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Greek General Secretariat for Research and Technology (GSRT), under the “IntWaste” project: Circular Economy—Intelligent Waste Management System.

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chryssolouris, G. *Manufacturing Systems: Theory and Practice*, 2nd ed.; Springer: New York, NY, USA, 2006.
2. Alexopoulos, K.; Sipsas, K.; Xanthakis, E.; Makris, S.; Mourtzis, D. An industrial Internet of things based platform for context-aware information services in manufacturing. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 1111–1123. [[CrossRef](#)]
3. Alexopoulos, K.; Nikolakis, N.; Chryssolouris, G. Digital Twin-Driven Supervised Machine Learning for the Development of Artificial Intelligence Applications in Manufacturing. *Int. J. Comput. Integr. Manuf.* **2020**, *33*, 429–439. [[CrossRef](#)]
4. Nti, I.K.; Adekoya, A.F.; Weyori, B.A.; Nyarko-Boateng, O. Applications of Artificial Intelligence in Engineering and Manufacturing: A Systematic Review. *J. Intell. Manuf.* **2022**, *33*, 1581–1601. [[CrossRef](#)]
5. Sipsas, K.; Alexopoulos, K.; Xanthakis, V.; Chryssolouris, G. Collaborative Maintenance in Flow-Line Manufacturing Environments: An Industry 4.0 Approach. *Procedia CIRP* **2016**, *55*, 236–241. [[CrossRef](#)]
6. Aivaliotis, P.; Anagiannis, I.; Nikolakis, N.; Alexopoulos, K.; Makris, S. Intelligent Waste Management System for Metalwork-Copper Industry. *Procedia CIRP* **2021**, *104*, 1571–1576. [[CrossRef](#)]

7. Hannan, M.A.; Al Mamun, M.A.; Hussain, A.; Basri, H.; Begum, R.A. A Review on Technologies and Their Usage in Solid Waste Monitoring and Management Systems: Issues and Challenges. *Waste Manag.* **2015**, *43*, 509–523. [[CrossRef](#)] [[PubMed](#)]
8. Mitra Tithi, D.; Chatterjee, P.; Chakrabarti, A. Smart Waste Monitoring Using Internet of Things. In *Data Management, Analytics and Innovation*; Sharma, N., Chakrabarti, A., Balas, V.E., Martinovic, J., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2021; Volume 1174, pp. 419–433.
9. Gopal Kirshna, S.; Manvi, S.S.; Bharti, P. Smart Waste Management Using Internet-of-Things (IoT). In Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies (ICCT), Chennai, India, 23–24 February 2017; pp. 199–203. [[CrossRef](#)]
10. Gopi, A.; Jacob, J.A.; Puthumana, R.M.; K, R.A.; S, K.; Manohar, B. IoT Based Smart Waste Management System. In Proceedings of the 2021 8th International Conference on Smart Computing and Communications (ICSCC), Kochi, India, 1–3 July 2021; IEEE: Kochi, Kerala, India, 2021; pp. 298–302.
11. Mastos, T.; Nizamis, A.; Vafeiadis, T.; Alexopoulos, N.; Ntinis, C.; Gkortzis, D.; Papadopoulos, A.; Ioannidis, D.; Tzouvaras, D. Industry 4.0 Sustainable Supply Chains: An Application of an IoT Enabled Scrap Metal Management Solution. *J. Clean. Prod.* **2020**, *269*, 122377. [[CrossRef](#)]
12. Papavasileiou, A.; Aivaliotis, P.; Aivaliotis, S.; Makris, S. An Optical System for Identifying and Classifying Defects of Metal Parts. *Int. J. Comput. Integr. Manuf.* **2022**, *35*, 326–340. [[CrossRef](#)]
13. Durga Prasad, P.; Muthuswamy, S.; Karumbu, P. Identification and Classification of Materials Using Machine Vision and Machine Learning in the Context of Industry 4.0. *J. Intell. Manuf.* **2020**, *31*, 1229–1241.
14. Swarit Anand, S.; Desai, K.A. Automated Surface Defect Detection Framework Using Machine Vision and Convolutional Neural Networks. *J. Intell. Manuf.* **2022**. [[CrossRef](#)]
15. Stavropoulos, P.; Papacharalampopoulos, A.; Petridis, D. A Vision-Based System for Real-Time Defect Detection: A Rubber Compound Part Case Study. *Procedia CIRP* **2020**, *93*, 1230–1235. [[CrossRef](#)]
16. Maher, A.; Hannan, M.A.; Begum, R.A.; Basri, H. Solid Waste Bin Level Detection Using Gray Level Co-Occurrence Matrix Feature Extraction Approach. *J. Environ. Manag.* **2012**, *104*, 9–18.
17. Martinez, P.; Mohsen, O.; Al-Hussein, M.; Ahmad, R. Vision-Based Automated Waste Audits: A Use Case from the Window Manufacturing Industry. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 7735–7749. [[CrossRef](#)]
18. Chaiwat, S.; Muangnak, N.; Pukdee, W. Designing of IoT-Based Smart Waste Sorting System with Image-Based Deep Learning Applications. In Proceedings of the 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Mai, Thailand, 19–22 March 2021; pp. 383–387. [[CrossRef](#)]
19. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]
20. Platform Industrie4.0-Specification Details of the Asset Administration Shell. Available online: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html (accessed on 10 December 2022).
21. Karen, S.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
22. Keras: Deep Learning for Humans. Available online: <https://github.com/keras-team/keras> (accessed on 31 January 2023).
23. Diederik, P.K.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
24. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
27. Manettas, C.; Nikolakis, N.; Alexopoulos, K. Synthetic Datasets for Deep Learning in Computer-Vision Assisted Tasks in Manufacturing. *Procedia CIRP* **2021**, *103*, 237–242. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.